



Implementation of OGC Web Services with MapServer

Karl Benedict, Sr. Research Scientist
Earth Data Analysis Center
University of New Mexico

Workshop presented at the
ESIP Federation Meeting
LDEO, July 2006



Workshop Goals

- Introduce the OGC service models sufficiently to productively outline their implementation
- Illustrate the implementation of three core OGC services
 - Web Map Services
 - Web Feature Services
 - Web Coverage Services



Workshop Outline

- Overview of the specific OGC web services to be addressed in the workshop
 - Web Map, Web Feature, and Web Coverage Services
- Overview of MapServer's configuration and deployment
- General strategy for implementation of OGC services in MapServer
- Specific implementation examples
 - WMS
 - WFS
 - WCS
- Other OGC specifications supported by MapServer

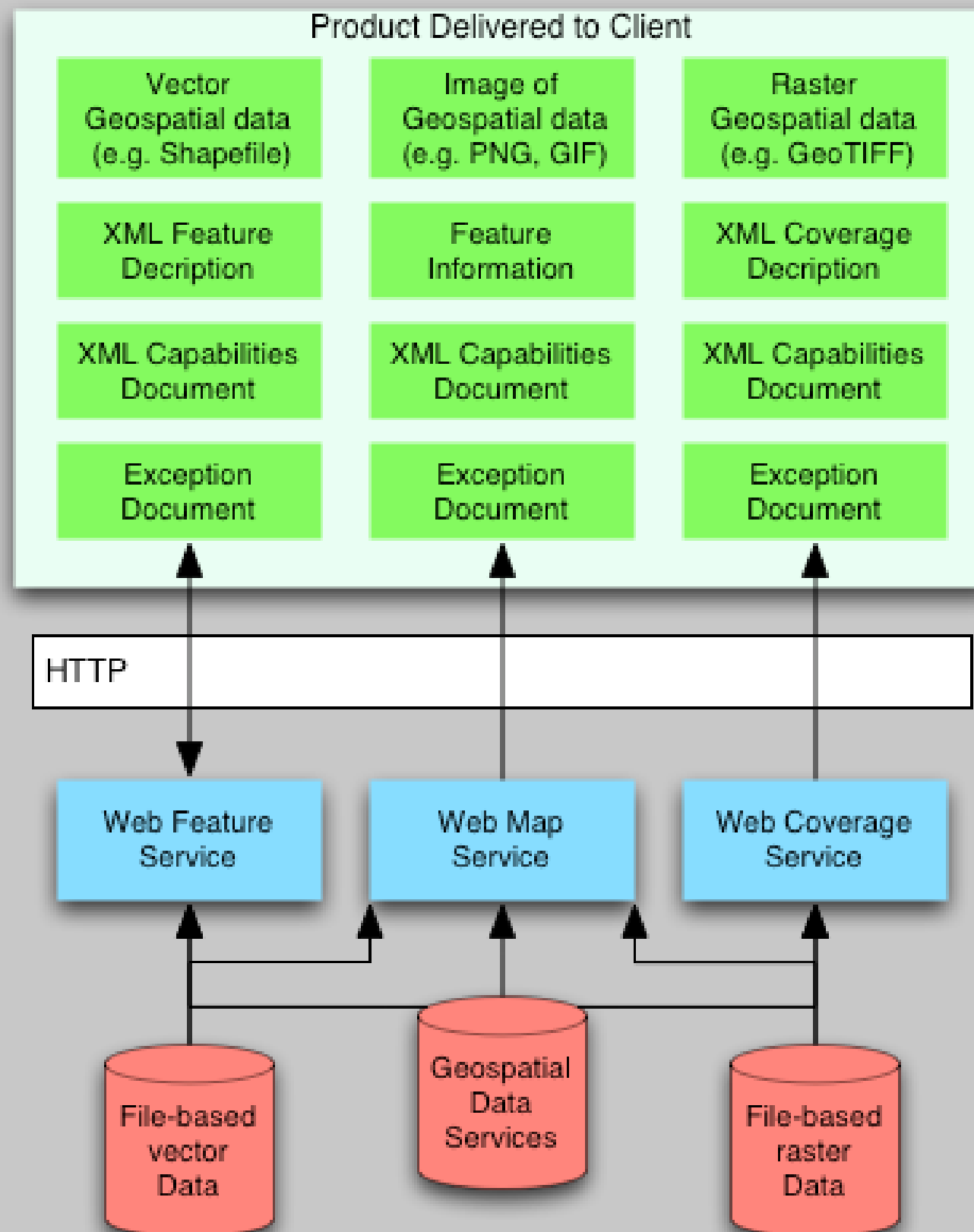


OGC Services Overview

There is support for several OGC services and specifications within MapServer. This presentation focusses on three:

- Web Map Services (images/maps)
- Web Feature Services (vector data)
- Web Coverage Services (raster data)

Comparison of Service Models





Functional Characteristics: WMS

- HTTP GET (required), HTTP POST (optional)
- Requests:
 - GetCapabilities
 - GetMap
 - GetFeatureInfo
- Returns
 - Mapped data
 - XML Capabilities Document, Feature Attributes

Request Parameters: WMS

| Parameter | Request | | |
|------------------|------------------------|---------------|-----------------------|
| | <i>GetCapabilities</i> | <i>GetMap</i> | <i>GetFeatureInfo</i> |
| VERSION | O | M | M |
| SERVICE | M | | |
| REQUEST | M | M | M |
| FORMAT | O | M | <i>GetMap</i> |
| UPDATESEQUENCE | O | | |
| LAYERS | | M | <i>GetMap</i> |
| STYLES | | M | <i>GetMap</i> |
| CRS | | M | <i>GetMap</i> |
| BBOX | | M | <i>GetMap</i> |
| WIDTH | | M | <i>GetMap</i> |
| HEIGHT | | M | <i>GetMap</i> |
| TRANSPARENT | | O | <i>GetMap</i> |
| BGCOLOR | | O | <i>GetMap</i> |
| EXCEPTIONS | | O | O |
| TIME | | O | <i>GetMap</i> |
| ELEVATION | | O | <i>GetMap</i> |
| other dimensions | | O | <i>GetMap</i> |
| QUERY_LAYERS | | | M |
| INFO_FORMAT | | | M |
| FEATURE_COUNT | | | O |
| I | | | M |
| J | | | M |

M=Mandatory, O=Optional



Functional Characteristics: WFS

- Either HTTP GET or POST required
- Requests
 - GetCapabilities
 - DescribeFeatureType
 - GetFeature/GetFeatureWithLock
 - GetGmlObject
 - LockFeature
 - Transaction
- Returns XML (GML), Capabilities, and Feature Data

Request Parameters: WFS

| Parameter | Request | | | | | |
|-------------------------------------|------------------------|----------------------------|--|---------------------|--------------------|--------------------|
| | <i>GetCapabilities</i> | <i>DescribeFeatureType</i> | <i>GetFeature & GetFeatureWithLock</i> | <i>GetGmlObject</i> | <i>LockFeature</i> | <i>Transaction</i> |
| VERSION | O | M | M | M | M | M |
| SERVICE | M | M | M | M | M | M |
| REQUEST | M | M | M | M | M | M |
| NAMESPACE | O | O | O | O | O | O |
| TYPENAME | | O | O/M | | O/M | O/M |
| OUTPUTFORMAT | | O | O | | | |
| RESULTTYPE | | | O | | | |
| PROPERTYNAME | | | O | | | |
| FEATUREVERSION | | | O | | | |
| MAXFEATURES | | | O | | | |
| EXPIRY | | | O | | O | |
| SRSNAME | | | O | | | |
| FEATUREID | | | O | | O | O |
| FILTER | | | O | | O | O |
| BBOX | | | O | | O | O |
| SORTBY | | | O | | | |
| TRAVERSEXLINKDEPTH | | | O ^a | M | | |
| TRAVERSEXLINKEXPIRY | | | O ^a | M | | |
| PROPTRAVXLINKDEPTH | | | O ^a | | | |
| PROPTRAVXLINKEXPIRY | | | O ^a | | | |
| GMLOBJECTID | | | | M | | |
| LOCKACTION | | | | | O | |
| OPERATION | | | | | | M |
| RELEASEACTION | | | | | | O |
| Vendor specific | O | O | O | O | O | O |
| M=Mandatory, O=Optional | | | | | | |
| ^a <i>GetFeature</i> only | | | | | | |



Functional Characteristics: WCS

- Either HTTP GET or POST required
- Requests
 - GetCapabilities
 - DescribeCoverage
 - GetCoverage
- Returns
 - Geospatial data for coverage
 - XML Capabilities

Request Parameters: WCS

| Parameter | Request | | |
|----------------|------------------------|-------------------------|--------------------|
| | <i>GetCapabilities</i> | <i>DescribeCoverage</i> | <i>GetCoverage</i> |
| REQUEST | M | M | M |
| VERSION | O | M | M |
| SERVICE | M | M | M |
| SECTION | O | | |
| UPDATESEQUENCE | O | | |
| COVERAGE | | O | M |
| CRS | | | M |
| RESPONSE_CRS | | | O |
| BBOX | | | M ^a |
| TIME | | | M ^a |
| PARAMETER | | | O |
| WIDTH | | | M ^b |
| HEIGHT | | | M ^b |
| DEPTH | | | M ^b |
| RESX | | | M ^b |
| RESY | | | M ^b |
| RESZ | | | M ^b |
| FORMAT | | | M |
| EXCEPTIONS | | | O |

M=Mandatory, O=Optional

^aEither BBOX or TIME is mandatory

^bEither WIDTH/HEIGHT/DEPTH or RESX/RESY/RESZ are mandatory



MapServer Configuration

- MapServer may be configured both as a client and as a server for the core OGC web service specifications:
 - Client: WMS, WFS
 - Server: WMS, WFS, WCS
- This presentation concentrates on server configurations in which the basic software requirements are the MapServer CGI, compiled with supporting required libraries
- A basic *map file* that provides the base information required by any MapServer implementation
- Enhancements to this *map file* that provide the additional information needed by MapServer to provide complete/compliant OGC WxS services.



Required MapServer Components

- The software requirements for MapServer's implementation of the OGC WxS specifications are typically met through the use of several open source programming libraries
 - Proj4 - geospatial coordinate transformation (reprojection)
 - GDAL/OGR - Raster and Vector data access, processing, and conversion libraries
 - GD - Graphics generation libraries
 - Xerces - XML libraries (for GML support)



Development of a Basic Map Service

- Install and configure the MapServer CGI - make sure that your version of MapServer supports the OGC specifications:

```
> mapserv -v./mapserv -v
```

```
MapServer version 4.8.3 OUTPUT=GIF OUTPUT=PNG
```

```
OUTPUT=JPEG OUTPUT=WBMP OUTPUT=PDF OUTPUT=SWF
```

```
OUTPUT=SVG SUPPORTS=PROJ SUPPORTS=FREETYPE
```

```
SUPPORTS=WMS_SERVER SUPPORTS=WMS_CLIENT
```

```
SUPPORTS=WFS_SERVER SUPPORTS=WFS_CLIENT
```

```
SUPPORTS=WCS_SERVER SUPPORTS=GEOS INPUT=EPPL7
```

```
INPUT=POSTGIS INPUT=OGR INPUT=GDAL INPUT=SHAPEFILE
```

- Acquire required data and metadata (particularly projection information)
- Develop a map file for the basic service (refer to Peri's previous talk)



Enabling OGC Services for a Map Service

- Compile and add required metadata content to the map file to enable the OGC services
- Map metadata - attributes that relate to the service as a whole
- Layer metadata - attributes that relate to a specific data 'layer' within the service
- Test by submitting *GetCapabilities* requests
- Test by submitting other data-related requests



General OGC Service Implementation Strategy

- If you are developing a pure OGC server (i.e. not developing a MapServer client interface based on HTML or MapScript), the most straightforward strategy is one of:
 - Bring together all needed data and metadata
 - Develop a simple mapfile that contains the minimum required information (including enabling metadata content) for the service
 - Test the provided information for completeness through issuing a *GetCapabilities* request to the server and reviewing the output capabilities document
 - Test the service with other supported requests (i.e. *GetMap*, *GetFeatureInfo*, etc.)



Implementation Examples - WMS

- WMS Requests Supportable by MapServer
 - GetCapabilities
 - GetMap
 - GetFeatureInfo
 - DescribeLayer
 - GetLegendGraphic



- **RGIS Previews**

- *GetCapabilities Request:*

`http://edacdata1.unm.edu/cgi-bin/mapserv?map=doqq05/
doqq05_demo.map&version=1.1.1&SERVICE=WMS&request=Get
Capabilities`

- *GetMap Request:*

`http://edacdata1.unm.edu/cgi-bin/mapserv?map=doqq05/
doqq05_demo.map&version=1.1.1&SERVICE=WMS&request=Get
Map&BBOX=-104.316429199742,36.184378915068,-104.24601
5087937,36.2531220444378&FORMAT=image/
png&STYLES=&LAYERS=doqq05&WIDTH=500&HEIGHT=500`



- **RGIS Previews**

- **AMIS Services**

- ***GetCapabilities Request:***

`http://amis.unm.edu/cgi-bin/mapserv?map=amis/
amis_demo.map&service=WMS&request=GetCapabilities`

- ***GetMap Request***

`http://amis.unm.edu/cgi-bin/mapserv?map=amis/
amis_demo.map&WMTVER=1.1.1&SERVICE=WMS&request=GetMap
&BBOX=-109.428,31.2527,-102.873,37.1093&FORMAT=image/
png&STYLES=&LAYERS=NMBoundary&WIDTH=500&HEIGHT=500`

`http://amis.unm.edu/cgi-bin/mapserv?map=amis/
amis_demo.map&WMTVER=1.1.1&SERVICE=WMS&request=GetMap
&BBOX=-109.428,31.2527,-102.873,37.1093&FORMAT=image/
png&STYLES=&LAYERS=NMBoundary,Landsat,Cities,Highways
&WIDTH=500&HEIGHT=500`



- PHAiRS Animation (Time-enabled WMS)

- *GetCapabilities Request:*

`http://phairs-devel.unm.edu:8080/cgi-bin/mapserv?
map=dream_p25_demo.map&VERSION=1.1.1&service=WMS&REQU
EST=GetCapabilities`

- *GetMap Request:*

`http://phairs-devel.unm.edu:8080/cgi-bin/mapserv?
map=dream_p25_demo.map&VERSION=1.1.1&service=WMS&REQU
EST=GetMap&BBox=-120.000,26.000,-97.000,44.000&SRS=EP
SG:
4326&Width=459&Height=360&Layers=GRASS_SHADED_RELIEF,
D121503_t01_pm25,usa_states,epa_airnow_complete&TIME=
2003-12-15T01`



Implementation Examples - WFS

- MapServer implemented requests:

- *GetCapabilities*

`http://amis.unm.edu/cgi-bin/mapserv?map=amis/
amis_demo.map&SERVICE=WFS&request=GetCapabilities`

- *GetFeatures*

`http://amis.unm.edu/cgi-bin/mapserv?map=amis/
amis_demo.map&version=1.0.0&SERVICE=WFS&request=GetFe
ature&typename=Highways,NMBoundary`



Implementation Examples - WCS

- **WCS Requests Supportable by MapServer**

- **GetCapabilities**

`http://edacdata1.unm.edu/cgi-bin/mapserv?map=doqq05/
doqq05_demo.map&SERVICE=WCS&request=GetCapabilities`

- **DescribeCoverage**

`http://edacdata1.unm.edu/cgi-bin/mapserv?map=doqq05/
doqq05_demo.map&version=1.0.0&SERVICE=WCS&request=Des
cribeCoverage`

- **GetCoverage**



Other Supported OGC Specifications

- Styled Layer Descriptors (SLD)
- Map Context
- Sensor Observation Service



- WMS Server How-To:
http://mapserver.gis.umn.edu/docs/howto/wms_server
- WFS Server How-To:
http://mapserver.gis.umn.edu/docs/howto/wfs_server
- WCS Server How-To:
http://mapserver.gis.umn.edu/docs/howto/wcs_server

- The NASA Earth Science Standards Process Group is seeking reviews and comments from the Earth Science community on the WMS 1.1.1 specification as a recommended community standard for NASA ES Data Systems
- The Standards Process working group web page:
<http://spg.gsfc.nasa.gov>
- The specific request for comment page:
<http://spg.gsfc.nasa.gov/rfc/ese-rfc-006>